Metadata Management on Data Processing in Data Lakes

Imen Megdiche, Franck Ravat, Yan Zhao

Metadata Management on Data Processing in Data Lakes

Imen MEGDICHE[1], Franck RAVAT[1], and Yan ZHAO[1,2]

[1] Institut de Recherche en Informatique de Toulouse, IRIT-CNRS (UMR 5505)
[2] Centre Hospitalier Universitaire (CHU) de Toulouse, Toulouse, France
Imen.Megdiche@irit.fr, Franck.Ravat@irit.fr, Yan.Zhao@irit.fr

Abstract. Data Lake (DL) is known as a Big Data analysis solution. A data lake stores not only data but also the processes that were carried out on these data. It is commonly agreed that data preparation/transformation takes most of the data analyst's time. To improve the efficiency of data processing in a DL, we propose a framework which includes a metadata model and algebraic transformation operations. The metadata model ensures the findability, accessibility, interoperability and reusability of data processes as well as data lineage of processes. Moreover, each process is described through a set of coarse-grained data transforming operations which can be applied to different types of datasets. We illustrate and validate our proposal with a real medical use case implementation.

Keywords: Data Lake · Data processing · Metadata Management

1 Introduction

Data Lake (DL) is a Big Data analysis solution that allows users to ingest raw data, store them in their native format, process these data upon usage, ensure the availability and accessibility of data for different users and apply governance to maintain the data quality, security and data life-cycle.

Data preparation is commonly considered as the most time consuming phase when analyzing data. Transformation processes, especially those in a DL, require a lot of effort because of (i) a great amount of different types of data (structured, semi-structured and unstructured) are ingested, (ii) various transforming operators can be carried out, for instance, consolidation, join, filtering, and (iii) different users are involved, such as BI (Business Intelligence) professionals, data scientists and data analysts. Users with different profiles apply different programs to prepare data by crossing various sources in a DL, in this paper, we use *data wrangler* to refer to these users.

To better govern a DL and to facilitate data preparation, metadata management is emphasized by many authors [1,3,4]. The integrated metadata dedicating to data processing allow data wranglers to find, access and reuse existing data transforming processes easier. Moreover, the source code and execution information allow users to update or adjust programs for further usages rapidly.

Today, different DL metadata solutions have been proposed or implemented in the academic and industrial world. However, most of the current solutions only focus on dataset metadata [1,3,4,2] without referring to process metadata. Moreover, some industrial solutions apply lineage metadata by only tracing source data and result data (Zaloni, Azure), but the process metadata is not specific and adapted enough for searching different types of processes efficiently by using the involved the operation or execution information.

At the aim of improving the reusability of data processes, it is better to describe a process through a sequence of generic operations than totally through free text. Hence, we define a framework that includes a metadata model in which processes are composed of a set of transformation operations. These later can be applied to different types of data. Moreover, the operations are presented with a controlled language to make the process interrogation easier. Note that we add the operation metadata for marking the main actions of a data process instead of translating each line of the transformation code (like ETL processes). Our framework have the following advantages: (i) The storage of processes and their metadata can facilitate data preparation by improving transparency and reusability of processes. (ii) The reliability of data is ensured by lineage metadata, users can verify the provenance of data to have more confidence and understanding of the data that they will use. (iii) Process metadata help data wranglers to find more relevant datasets, for instance, datasets that are generated by the same process.

At the aim of proposing a metadata management focusing on data processing for data lakes, our paper is threefold. In section 2, we introduce related work on metadata dedicated to data processing. In section 3, we propose a data lake metadata model for data processes with a minimal core of transforming operations with illustrations on our motivating example. Finally, in section 4, we present an implementation of our model and we validate it by technical aspects.

2 Related Work and Motivating Example

To the best of our knowledge, there are a few works [5,11] in the literature presenting metadata on data processing in the context of Big Data (contrarily to the multitude of works in the data warehousing / ETL processes [12,9])

The authors of [11] introduced a metadata system for primary care big data to control the process of transformation and analysis. The system adds six elements of metadata to the Primary Care Data Quality (PCDQ) renal program: study/audit name, queries of data extraction, data collection number, data type, repeat number and a processing suffix. As described in this approach, we observe that there is a focus on quality aspects which does not cover all the transformations and problems that we find in DL. Moreover, in this work there is not a generic metadata model presented.

The authors of [5] define a metadata schema describing data preparation tasks in the context of data mining. The system aims to automate data preparation by identifying its requirements which are classified into eight categories: objective, output, definition, control, flow, content, composition and execution.

This approach focuses on data preparation metadata for data mining and the metadata model specify all tasks of a data process. Showing all the details of data processes can facilitate the comprehension of process, but in the context of data lakes, it is too heavy for both metadata extraction and metadata searching.

Contrary to the previous work, as in DL, we pull together all type of users (BI professionals, data statisticians, data analysts, data engineers...), we need to have a generic model covering all types of transformations for each kind of data. The DL metadata system must integrate essential activities of data processing to ensure that all DL users can efficiently find and reuse existing processes.

Motivating Example. In order to exemplify our metadata model, we rely on an example throughout the paper. The motivated case is based on a feedback experience carried out on the DL of the University Teaching Hospital (CHU) of Toulouse. The purpose of this DL is to gather different types of medical data from the complex information system that contains more than 100 large or small datasets for future analysis. For the scenario, we keep only two projects in the scope of the data in order to validate our solution. The Fig. 1 shows the workflow of data processing for these projects.

EHDEN Project uses the electronic health record (EHR) dataset as the data source and creates the OMOP dataset accordingly for collaborative analysis.

- Clinical data are extracted and fed by several steps. SP1.1 extracts different subjects (e.g. patient, medical staff, diagnosis) of clinical data. During SP1.2, the extracted data are validated manually by EHR experts and doctors then stored in CSV files. SP1.3 transforms validated data to OMOP CDM. When the OMOP format clinical data are validated during SP1.4, they are used to feed the OMOP clinical tables during the SP1.5.
- Terminologies used in the EHR need to be mapped to OMOP standardized vocabularies. Firstly, during SP2.1, each terminology (e.g. for diagnosis, medication, procedure) used in the EHR are extracted and stored in CSV files. SP2.2 aims to validate the extracted data. SP2.3 concerns to map different terminologies to OMOP standardized vocabularies. SP2.4 needs to be done by doctors to validate the mapping. When all the terminologies are mapped, relative OMOP tables are fed.

EBERS Project aims to analyze all textual medical reports stored in the CHU database using NLP (Natural Language Processing) techniques. The current database stores medical report information in different tables. Therefore, to prepare NLP, we need to firstly reconstruct medical reports with the data stored in the EHR and apply further transformations.

3 An Extended Metadata Model for Data Processing

3.1 Meta-model

To ensure the findability, accessibility, interoperability and reuability of data processing, we propose that data processing metadata include four targets:

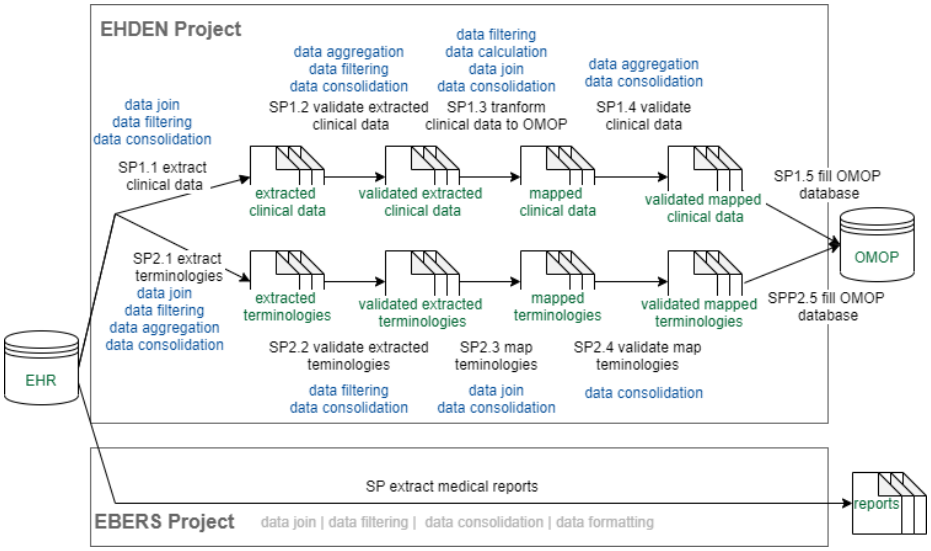


Fig. 1. Motivating Example

- *Process Characteristics* introduces basic information about a process. These metadata answer the question: "Who did what when".
- *Process Definition* explains why a process is created, what is the context, what is the objective. They specify the meaning of data processes.
- *Technical information* includes process code and execution information. The technical information helps users to know the deployment environment. The source code also allow users to modify, update or reuse a process.
- *Process content* concerns coarse-grained transformation operators dedicating to qualify data processing.

Firstly, we have proposed a first version of a generic and extensible metadata model that mainly focuses on datasets in [8]. In this paper, we answer the previous requirements (4 targets), so that we propose an extension of the model with detailed metadata about data processing.

From the modeling point of view (see Fig. 2), the metadata associated with the previous 4 targets are modeled through 6 object classes. (i) Process characteristics includes source and target dataset(s), name, creation date and the user who works on the process. This information is stored in 3 different classes *DatalakeDataset, Process, User* (marked in blue). (ii) Each procces is defined by a description and a set of keywords (marked in yellow). (iii) Technical information contains the source code and execution details (marked in green). And (vi) process content includes a set of coarse grained operations (marked in red).

The process content describes the data processing operations at a high level. The objective of operation metadata is not to represent all the details of the

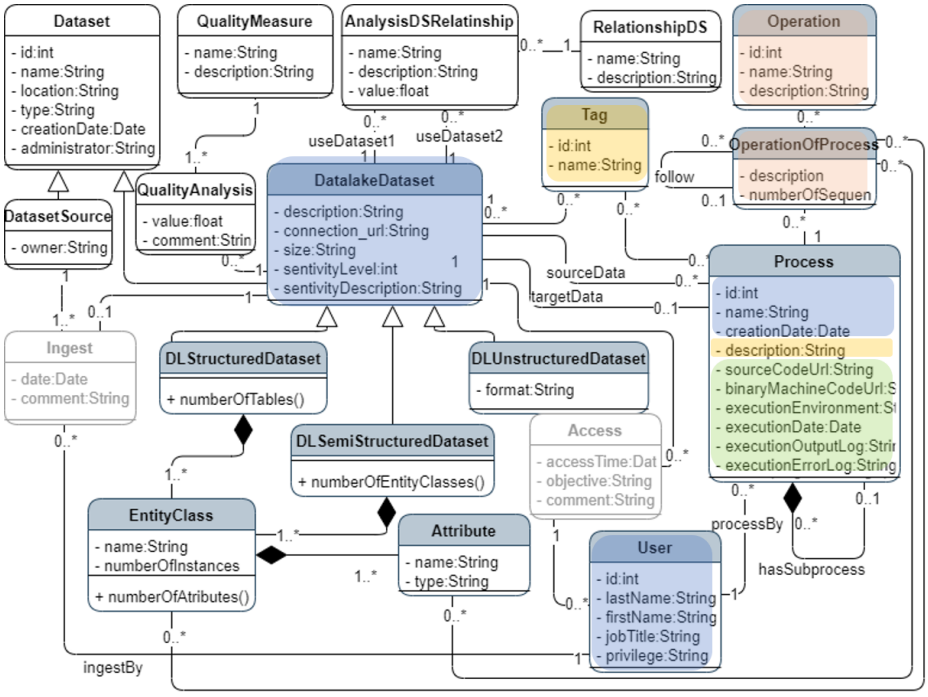


Fig. 2. Metadata conceptual model

source code but to help DL users to get a glimpse the main activities of the processes. For instance, one project needs to create a dataset DS1 by using two sources DS2 and DS3. Instead of writing source code directly, the user can check if there are already some processes which use DS2 and DS3 as sources and are marked with 'merge' or 'join' operations.

3.2 Operations

To complete the meta-model and facilitate the feeding, we propose to describe data processes through a set of coarse-grained operations. Each operation is defined as $DS_{output} \leftarrow OP((DS_{input1}, [DS_{input2}...DS_{inputN}]), ARG)$, where:

- OP is the data transforming operation.
- $DS_{input1}, [DS_{input2}]...[DS_{inputN}]$ is the source dataset(s) of the operation.
- ARG is the argument of the operation, it can be a condition or a function of an operation.
- DS_{output} is the result of the operation.

In order to make the operators generically compatible with classical ETL operations [7,10,13] as well as data preparation operators for data mining [6], or

specific practices like reducing data or discretizing data [14], the predefined list is $OP \in$ {FL, FMT, AGGR, CALC, CNSLD, MERGE, JOIN}. We introduce each operation with examples from the projects in 1:

Filtering (FL) has the objective of choosing a subset of data from the data source with conditions: $FL(DS_{input}, [COND])$. The conditions concerns the selection of attributes and/or instances. ***Example.*** During the SP2.1, all the diagnoses used by the CHU of Toulouse need to be extracted. However, in the EHR, different versions of diagnoses in French and German are stored. Therefore, the FL operation is needed: $Ex1 \leftarrow FL(diagtype, (cc =' FR'))$.

Formatting (FMT) has the objective to transform a dataset from their native format into a predefined format: $(DS_{input}, [targetFormat])$. ***Example.*** The medical reports in the EHR are stored in the form of relational data (title, type, reporter, text, etc.), while we need to extract reports which has a letter format, therefore, the structured data need to be transformed to unstructured: $Ex5 \leftarrow FMT(report, unstructured)$.

Aggregation (AGGR) has the objective to gather data and present the data in a summary form: $AGGR(DS_{input}, [ATTR], [FUNC])$, where ATTR is a set of attributes to group by and FUNC concerns the aggregate functions. ***Example.*** To facilitate validating extracted terminologies, for instance, all the diagnoses, we need to count the diagnosed frequency of diagnoses: $Ex3 \leftarrow AGGR($ $JOIN((Ex2, stay\_diag), (Ex2.diagid = stay\_diag.diagid)), diagid, count(diagid))$.

Calculation (CALC) has the objective to calculate additional data with existing information: $CALC(DS_{input}, FUNC)$. FUNC may contain parameters like input attributes, output attribute and associated calculating function, for instance, mathematical, date or user defined functions. ***Example.*** In the EHR, patients' birth date information is stored in the format of DD/MM/YYYY, while to count the distribution of patients' age, calculation is needed: $Ex6 \leftarrow$ $CLAC(patient, (today() - patient.birthdate))$.

Consolidation (CNSLD) concerns converting, correcting or protecting data: $CNSLD(DS_{input}, [ATTR], [COND]$. (i) Data conversion concerns the modification of data format, the updating of data type, the splitting of data, the data combination, the data normalization and the value generator. (ii) Data correction concerns the correction of missing or incorrect data. (iii) Data protection concerns limiting the authorized access by data encryption or anonymizing data to ensure the privacy of personal data. Data protection can be applied to all the structural types of datasets. Anonymization, data pseudonymization concerns the encryption or erasure of identifiable personal information to ensure the privacy protection. ***Example.*** In the EHR, the free text of medical reports is store in base64, to apply NLP processes, it needs to be decoded in UTF-8: $Ex4 \leftarrow CNSLD(report, (text), convert(UTF - 8))$.

Merging (MERGE) has the objective to combine datasets that have compatible elements: $MERGE((DS_{input1}, DS_{input2}...DS_{inputN}), [COND])$. ***Example.*** three classifications are already validated and mapped to OMOP format, they need to be merged: $Ex6 \leftarrow MERGE(diag, procedure, medication)$.

Operation	Structured DS	Semi-structured DS	Unstructured DS
Filtering	**FL(DS_{input}, [COND])** FL(DS, n / n%) *randomly select n lines* FL(DS, (attr1, attr2…attrn)) *select attributes (projection)* FL(DS, (attrx<m)) *select by value of attributes (selection of instances)*	**FL(DS_{input}, [COND])** FL(DS, n / n%) *randomly select n lines* FL(DS, (attr1, attr3…attrn)) *select attributes* FL(DS, (attrx<m)) *select by value of attributes*	**FL(DS_{input}, [COND])** FL(DS, n / n%) *randomly select n parts* FL(DS, Cond) *select by UDF*
Formatting	**FMT(DS_{input}, [targetFormat])** *targetFormat: Structured/semi-structured/unstructured*	**FMT(DS_{input}, [targetFormat])** *targetFormat: Structured/semi-structured/unstructured*	**FMT(DS_{input}, [targetFormat])** *targetFormat: Structured/semi-structured/unstructured*
Aggregation	**AGGR(DS_{input}, [ATTR], [FUNC])** AGGR(DS, (attr1, attr2,...attrn), avg / count / min / max / sum / UDF (attr))	**AGGR(DS_{input}, [ATTR], [FUNC])** AGGR(DS, (attr1, attr2,...attrn), avg / count / min / max / sum / UDF (attr))	-
Calculation	**CALC(DS_{input}, FUNC)** *function: mathematical, date, UDF*	**CALC(DS_{input}, FUNC)** *function: mathematical, date, UDF*	-
Consolidation	**CNSLD(DS_{input}, [ATTR], [COND])** *Convert data format, Update data type, Splitting data, Combining data, Data standardization, Correct missing data, Correct incorrect data, Encrypting data, Anonymizing data, Pseudonymizing data*	**CNSLD(DS_{input}, [ATTR], [COND])** *Convert data format, Update data type, Splitting data, Combining data, Data standardization , Correct missing data, Correct incorrect data, Encrypting data, Anonymizing data, Pseudonymizing data*	**CNSLD(DS_{input}, [ATTR], [COND])** *Correct incorrect data, Encrypting data, Anonymizing data, Pseudonymizing data*
Merging	**MERGE((DS_{input1},DS_{input2} … DS_{inputN}), [COND])** MERGE(DS1, DS2…DSn, n% / total) *UNION in SQL*	**MERGE((DS_{input1},DS_{input2} … DS_{inputN}), [COND])** MERGE(DS1, DS2…DSn, n% / total) *Delimited text file: Adding data by rows* *NoSQL/XML/JSON: Adding data by objects/nodes*	**MERGE((DS_{input1},DS_{input2} … DS_{inputN}), [COND])** MERGE(DS1, DS2…DSn, n% / total) *Adding data by paragraphs/pages*
Join	**JOIN((DS_{input1}, DS_{input2}), [COND])** JOIN((DS1, DS2), DS1.attr1 = DS2.attr1))	**JOIN((DS_{input1}, DS_{input2}), [COND])** JOIN((DS1, DS2), DS1.attr1 = DS2.attr1))	-

Fig. 3. Coarse-Grained operations

Join (JOIN) has the objective to combine different data sources with common values, it can be applied on (semi-)structured data: $JOIN((DS_{input1}, DS_{input2}), [COND])$, where COND is the condition on the common value. ***Example.*** To extract all the diagnoses in French, the tables of diagnosis and types need to be joined: $Ex2 \leftarrow JOIN((diag, diagtype), (diag.typeid = diagtype.typeid))$.

All these coarse grained operations can be applied to different structural types of data (see Fig. 3). FL, FMT, AGGR, CALC, CNSLD are unary operations which can be applied on a single dataset. MERGE and JOIN are binary operations which can be applied on multiple datasets, but these datasets should have the same structural type, if it is not the case, the FMT is used to convert dataset to the required structural type. The FMT is also the only operation that changes data format in our list.

3.3 Application

To better introduce the operations, we present an application of a sub-process in Fig. 1. The SP2.1 aims to extract different terminologies from the EHR database. The drug classification used at Toulouse CHU is UCD10. UCD10 is a national classification in France of all the approved drugs. To obtain the stored UCD10 and facilitate the validation phase, we extract all the drugs with their number of prescription times. The useful information is stored in 4 different tables and to respect the copy right of the database schema, we do not show the full name of tables or attributes.

$DS_{extracted\_drug} \leftarrow$ *CNSLD(AGGR(FL(JOIN((JOIN((JOIN((JOIN((presc, medic), (presc.drugid = medic.drugid)), medpra), (presc.drugid = medpra.drugid(+))), medbook), (presc.drugid = medbook.medid)), art), (mdbook.artid = art.artid)), (presc.cancel = ")), (presc.drugid, medic.meducd, art.ucd, presc.drugname, medic.meddrugname, art.artdrugname), COUNT(presc.drugid)),check_missing_value(presc.drugid))*

Regarding the example, for the process $DS_{extracted\_drug}$, 7 objects of OperationOfProcess are created. Each of the objects represents one operation with its argument. And these objects are linked to 4 different objects of Operations: CNSLD, AGGR, FL and JOIN.

4 Exploitation of the Metadata

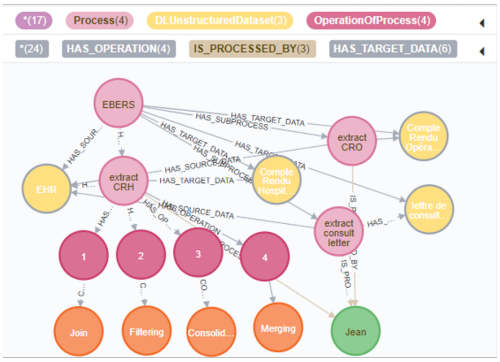
To illustrate that our metadata model of data processing can help users find the relevant processes or datasets to improve the efficiency and efficacy of data preparation, we have implemented a graph database by Neo4j. The choice of a graph database is motivated by the scalability and flexibility as well as the interconnections of this NoSql storage system. We choose the Neo4j platform for its maturity.

The implemented database of the motivate case is composed of more than 1300 nodes and 1600 relationships. There are 10 types of nodes: DLStructuredDataset, DLSemiStructuredDataset, DLUnstructuredDataset, Process, OperationOfProcess, Operation, Keyword, User. Due to limited space, the schema of the database is presented on Github[3]. According to the schema, concerning the motivate case, different structural types of data are stored, different processes can be applied to these data and a process can have sub processes. A process is composed of a set of operation and it is carried out by a user.

We emphasize that the process metadata should at least help users on the following stages: (i) **When creating a new dataset from existing source**, data wranglers can find all other datasets created from the same source and their corresponding processes. (ii) **When working on existing dataset**, data wranglers can find the history of the different types processes that were executed on this dataset. (iii) **While manipulating process**, data wranglers need to reuse a data process or if they want to modify or update a process for further use, they can find all the source code and execution information.

To validate our implementation and present its application on data preparation, we introduce one example on searching data processing metadata in this paper (there is another example available on the Github project) : A head trauma doctor wants to analyze all the medical history of his patients to have more information and to improve the effectiveness of his medical treatment. A part of his project concerns the analysis of various medical reports. He annotated a few keywords of three types of reports (paper version): hospitalization reports, neuropsychological assessments and neuropsychological examinations. The team

[3] `https://github.com/yanzhao-irit/metadata_management_on_data_processing_in_data_lakes`




```
MATCH
  (p:Process)-[:HAS_TARGET_DATA]-(d),
  (p)-[:HAS_OPERATION]->(op:OperationOfProcess),
  (op)-[:CONCERNS_OPERATION]->(o:Operation),
  (p)-[:HAS_SUBPROCESS]-(p2:Process),
  (p)-[:IS_PROCESSED_BY]-(u),
  (p2)-[*1]-(n)
WHERE
  toLower(d.name) = 'compte rendu hospitalisation'
  OR toLower(d.name) = 'evaluation neuropsychologique'
  OR toLower(d.name) = 'examen neuropsychologique'
RETURN p,d,op,o,p2,u,n
```

Fig. 4. Query and result of example

who works on the project needs to extract the electronic version of these reports from the EHR, annotate what he marked and analyze these reports. The first step of this project is to extract the three types of reports. For the reason that the EHR database is not well documented, the team does not know where to find the corresponding data and how to restructure the reports. Therefore, they use the metadata system to check if the three types of reports are already extracted. The used query is presented below and the result is in Fig. 4.

With the result, the team discovers that hospitalization reports are already extracted for the EBERS project. Although they cannot find the other two types of reports, after studying the queries, they know that all reports data are stored in three tables. They also contact the person who worked on EBERS to request more experiences on extracting medical reports. The effectiveness of their work is much improved by the experience from the project EBERS.

5 Conclusion and Future Work

To the best of our knowledge, there is no solution which can take advantage of the existing processes in a data lake by improving the findability, accessibility, interoperability, and reusability. To better use a data lake, in this paper, we proposed a metadata model including metadata on data processing which can help users to find or even reuse certain processes to make the data transformations more efficient. The introduced process metadata contain operation metadata which are classified by coarse grain into seven categories. These operations can be described by a controlled language. We implement the metadata model in a graph database with Neo4j and validate it.

The graph database of metadata is the basic building block for a metadata management system for a data lake. For future work, we have to proceed on two fronts: (i) The automatic extraction of metadata. Although some metadata have to be entered manually, for instance, the name, description, keywords of pro-

cesses, automatic metadata extraction is always essential which helps to reduce the time to metadata management and avoid possible manual entry errors; (ii) A graphical and ergonomic interface of metadata management system. There is a need to provide an interface that allows data lake users to easily search without writing complicated requests.

References

1. Alserafi, A., Abelló, A., Romero, O., Calders, T.: Towards information profiling: data lake content metadata management. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). pp. 178–185. IEEE (2016)
2. C. Quix, R. Hai, I.V.: Metadata extraction and management in data lakes with gemms. Complex Systems Informatics and Modeling Quarterly pp. 67–83 (12 2016)
3. Diamantini, C., Giudice, P.L., Musarella, L., Potena, D., Storti, E., Ursino, D.: An approach to extracting thematic views from highly heterogeneous sources of a data lake. In: Atti del Ventiseiesimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD'18) (2018)
4. Halevy, A., Korn, F., Noy, N.F., Olston, C., Polyzotis, N., Roy, S., Whang, S.E.: Goods: Organizing google's datasets. In: Proceedings of the 2016 International Conference on Management of Data. pp. 795–806. ACM (2016)
5. Hidalgo, M., Menasalvas, E., Eibe, S.: Definition of a metadata schema for describing data preparation tasks. In: Proceedings of the ECML/PKDD 2009 Workshop on 3rd generation Data Mining (SoKD 2009). pp. 64–75 (2009)
6. Jin, Z., Anderson, M.R., Cafarella, M., Jagadish, H.: Foofah: Transforming data by example. In: Proceedings of the 2017 ACM International Conference on Management of Data. pp. 683–698. ACM (2017)
7. Poole, J.: The common warehouse metamodel as a foundation for active object models in the data warehouse environment. In: ECOOP 2000 workshop on Metadata and Active Object-Model Pattern Mining-Cannes, France (2000)
8. Ravat, F., Zhao, Y.: Metadata management for data lakes. In: European Conference on Advances in Databases and Information Systems. pp. 37–44. Springer International Publishing (2019)
9. Simitsis, A., Vassiliadis, P., Dayal, U., Karagiannis, A., Tziovara, V.: Benchmarking etl workflows. In: Technology Conference on Performance Evaluation and Benchmarking. pp. 199–220. Springer (2009)
10. Trujillo, J., Luján-Mora, S.: A uml based approach for modeling etl processes in data warehouses. In: International Conference on Conceptual Modeling. pp. 307–320. Springer (2003)
11. VanVlymen, J., de Lusignan, S.: A system of metadata to control the process of query, aggregating, cleaning and analysing large datasets of primary care data. Journal of Innovation in Health Informatics **13**(4), 281–291 (2005)
12. Vassiliadis, P., Simitsis, A., Baikousi, E.: A taxonomy of etl activities. In: Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP. pp. 25–32 (2009)
13. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for etl processes. In: Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP. pp. 14–21. ACM (2002)
14. Zhang, S., Zhang, C., Yang, Q.: Data preparation for data mining. Applied artificial intelligence **17**(5-6), 375–381 (2003)